

Advanced IT Auditing: Determining Quality When the QMS Doesn't.

Barry McManus,
Principal Consultant

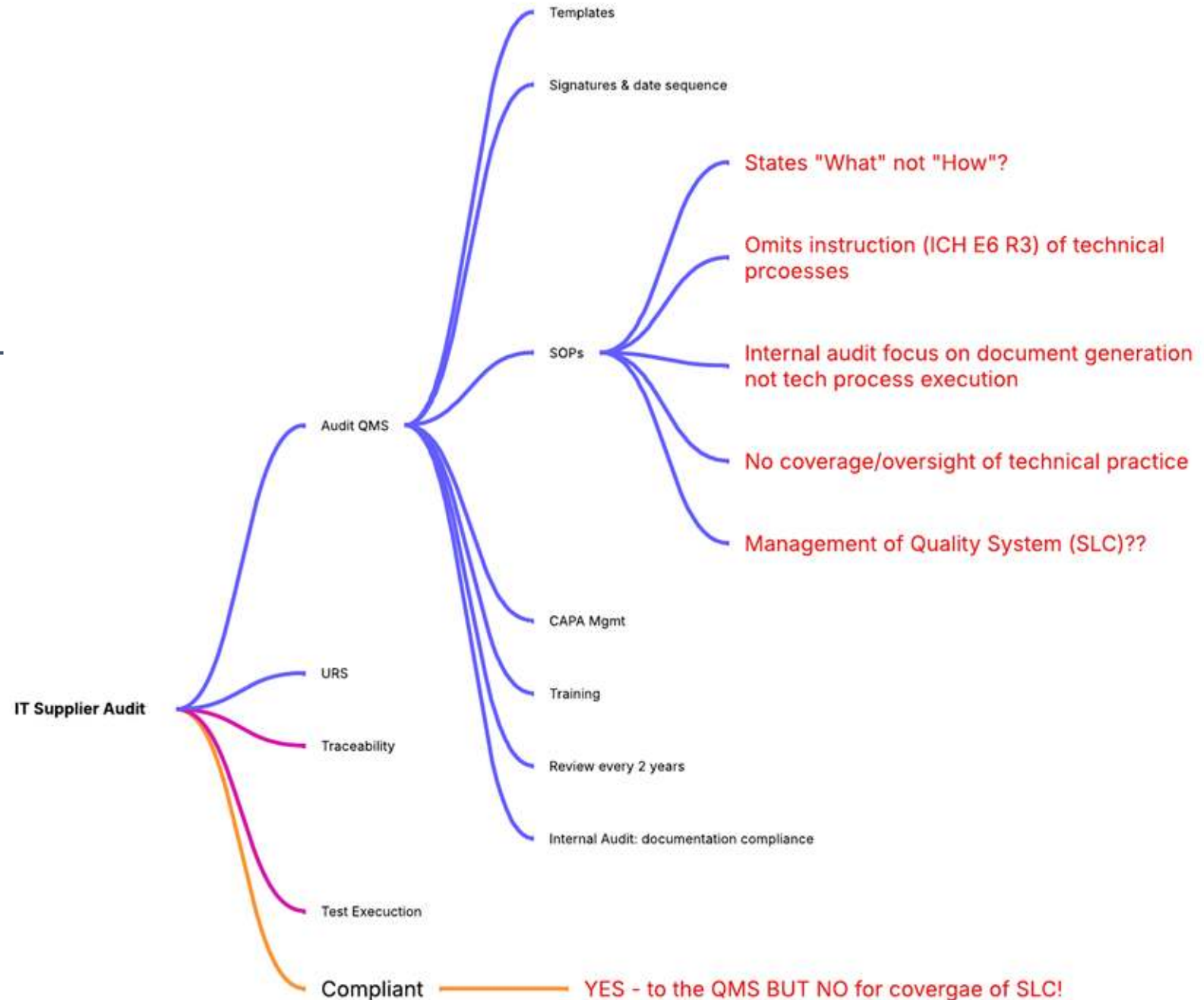


Hugh O'Neill,
VP Operations & Quality

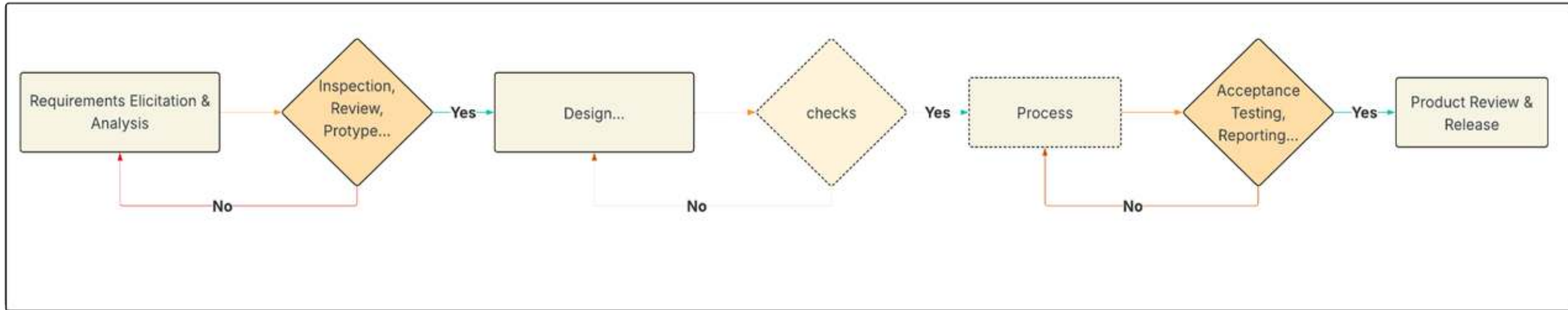


Agenda:

- The production line analogy.
- The challenge.
- Dealing with the problem: 2 techniques.
- 3rd Technique: measurement.
- Regulatory thoughts on measurement.
- QMS assessment by measurement.
- Case study: supplier' measurement journey.
- Summary.

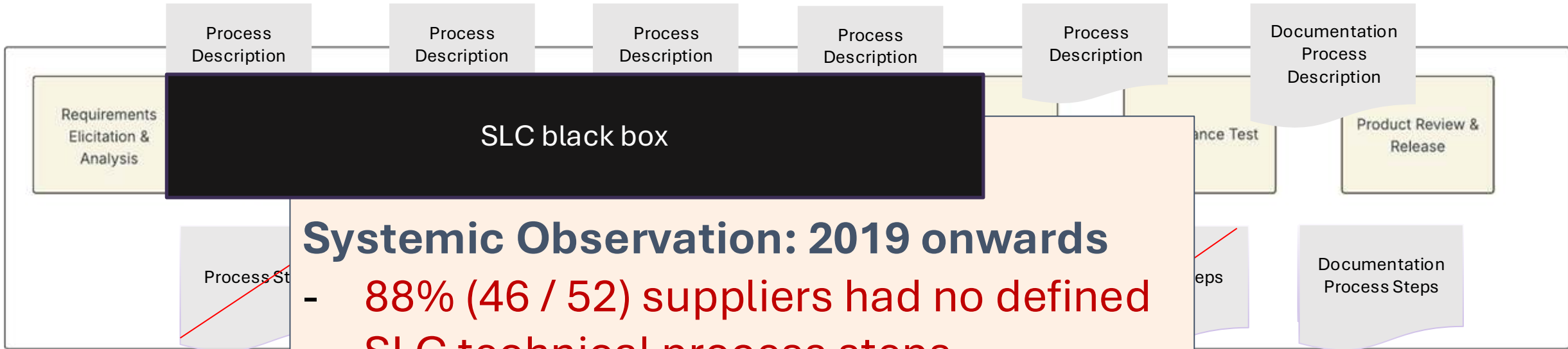


The Quality System is a Production Line









- Manufacturing = processes -> quality product -> accepted.
- Manufacturing = Need process **instruction** = **uniformity** of **execution** -> minimum quality.
- Use (Plan) **Do-Check-Act** = minimise error + remediation costs = productivity/revenue.
- SLC = software (manufacturing) production line -> but more **complex**.
- Same mindset... sequence of processes + Do-Check-Act per process.
- Audit: **evaluate** -> compliance (regulatory) + non conformance (to process).
- Audit: **evaluate** -> effectiveness (does it work or just exist) + continuous improvement.

The Challenge: Does the QMS have oversight?



Systemic Observation: 2019 onwards

- 88% (46 / 52) suppliers had no defined SLC technical process steps.
- Documentation compliance.
- **But QMS process gaps.**

-  SOP describe
-  QMS describe
-  Work Instruct
-  Staff trained
-  Documentation compliance.
-  Process reflects the QMS.

-  Software Product Quality improvement?
-  QMS isn't managing the production line.

Advanced IT Audit Technique: “2ndary” Information & Walkthrough

‘Prove’ we have good quality?... Our customers love us.. We’re #1!

... oversight of software pro... (to detect Data Integrity issues).

Ha, ha... poor Barry – that’s a traditional auditing question. We’re agile!

Secondary Information.

Everyone knows that all software has issues.... How can you expect us to catch them all..?

- SOPs - state the “what”
- WI’s - had no instructions - the “how”.
- ... maintained level of instructions – BUT outside of QMS & audit scope.
- ... defined to facilitate independent QA oversight.

Technical Walkthrough revealed process execution and check...

- Requirements: Jira #1527 - no statements. No objective measur...
- Risk Assessment: Missing from requirements.
- Design: requested on day 1, delivered on day 3. Auditee generates...
- Code (...)
- Code (...)
- Unit Test (...)
-

You don’t understand.... You don’t have to validate, we validate... you just use our system.

Eh?.. No-one else asks for this.... We might consider it if other auditors raise this point.

Our WI’s lack instruction? ...Our developers know what they are doing!

... SOC2 Auditors’ did not review software compliance or effectiveness thereof.

3rd Technique: Use Non-Conformance of Software Product

ICH Q8 – Quality is ‘fitness... of intended use’

IF a **defect** is a **non-conformity** to a requirement (intended use)

THEN we can use **defect** as a measure to **assess** the **quality** of:

The *software product*

&

The *QMS that manages the quality processes (that build s/w quality)*.

3rd Technique: Measurement – Regulatory View.

- EMA GVP 2011: **Quality ... can be measured**. Measuring .. the required degree of quality necessitates *pre-defined quality objectives*.
- EU Annex 11 2011: ...formal **assessment** and reporting **of quality** & performance **measures** for **all .. lifecycle stages**.
- FDA 2002: ... V&V are difficult... can't test for ever.. Hard to know how much evidence is enough... **Measures** such as *defects found... estimates of defects remaining...* are used to develop an acceptable level of confidence.
- IEEE 1061: .. A **quantitative measure** of the degree to which software possesses a given attribute which affects its **(of) quality**.

3rd Technique: Measurement - Standards

- ISO9001:1994: Clause 4.9 (d) **monitoring & control of process parameters** and **product characteristics** (*infers* measure of defect metrics).
- ISO9001:2000: Clause 8: 8.2.3 Monitoring and measurement of processes; 8.2.5 Monitoring and measurement of product; 8.4 **Analysis of Data** – (c) *characteristics and trends of processes and products including opportunities for preventive action.*
- ISO9001:2008: Same as for 2000.
- ISO9001:2015: Clause 9.1.3 Analysis & Evaluation – ... analyse appropriate data ... to **evaluate**
(a) **conformity of products**.... (b) customer satisfaction.... (c) **effectiveness of the QMS**.. (d) ...effective planning... (e) ... actions to address risks & opportunities... (f) performance of external suppliers... (g) need to improve the QMS

3rd Technique: Measurement examples

(non conformance) Defect Cost & Effort | Defect RCA | Defects Not fixed:

- Defects **per Release**: Increasing, decreasing, static..
- Defect **Location**: Features where defects arise. (Least burdensome approach)
- Defect **Severity**: severity of defects within a release. (1960s)
- Defect **Type**: Data, boundary values, performance, memory, password reset, error handling...

Defect Removal Efficiency (DRE): percentage of defects found and repaired prior to release to the customer. (IBM 1973, FDA 2002 *estimates of defects remaining*)

- If your supplier finds 90 defects pre release and you find 10 (5 in validation and 5 in production) - Supplier process: 90% DRE (90/100)
- [Collaborative approach] Supplier & Validation: 95% DRE (95 / 100)
- **Predictability**: IF we use same version of processes/personnel/tools THEN

We can **estimate** ^(FDA 2002) 5% of defects will 'slip' into production for next release.

3rd Technique: How metrics can reveal process effectiveness



Product quality across test releases

FIGURE 16. DEFECTS FOUND PER SYSTEM TEST ITERATIONS

	HO1	HO2	HO3	HO4	HO5	HO6	TOTAL DEFECTS
Defect Totals	62	15	1	2	3	3	86
DDP %	-	80.5	98.72	97.5	96.39	96.51	-

Product quality across years

FIGURE 17. DEFECT TOTALS ACROSS 30 MONTHS

SYSTEM TEST DEFECTS PER YEAR		
2022	2023	2024 (6 months)
162	110	43

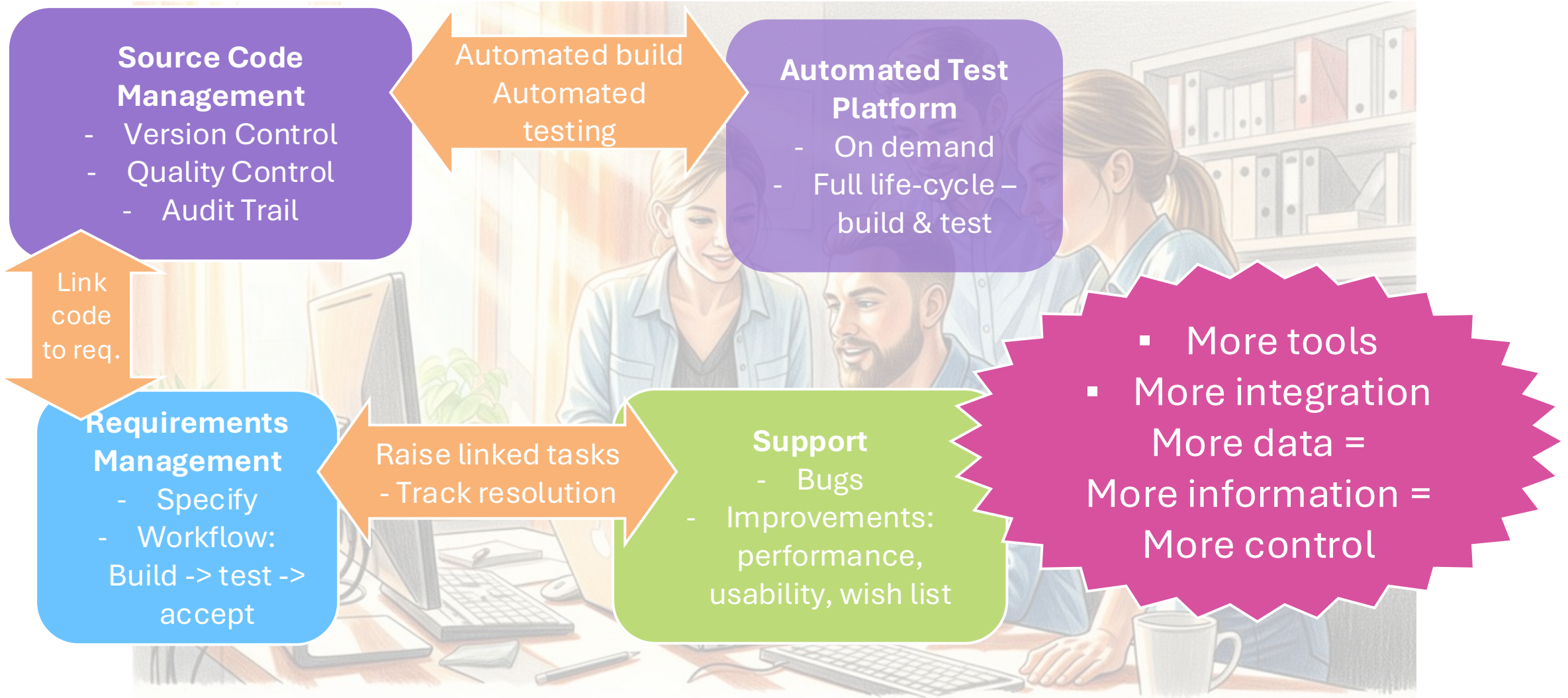
Product quality across features/year

FIGURE 18. SIMPLE DEFECT TRENDING TO INDICATE FEATURE QUALITY ACROSS 30 MONTHS

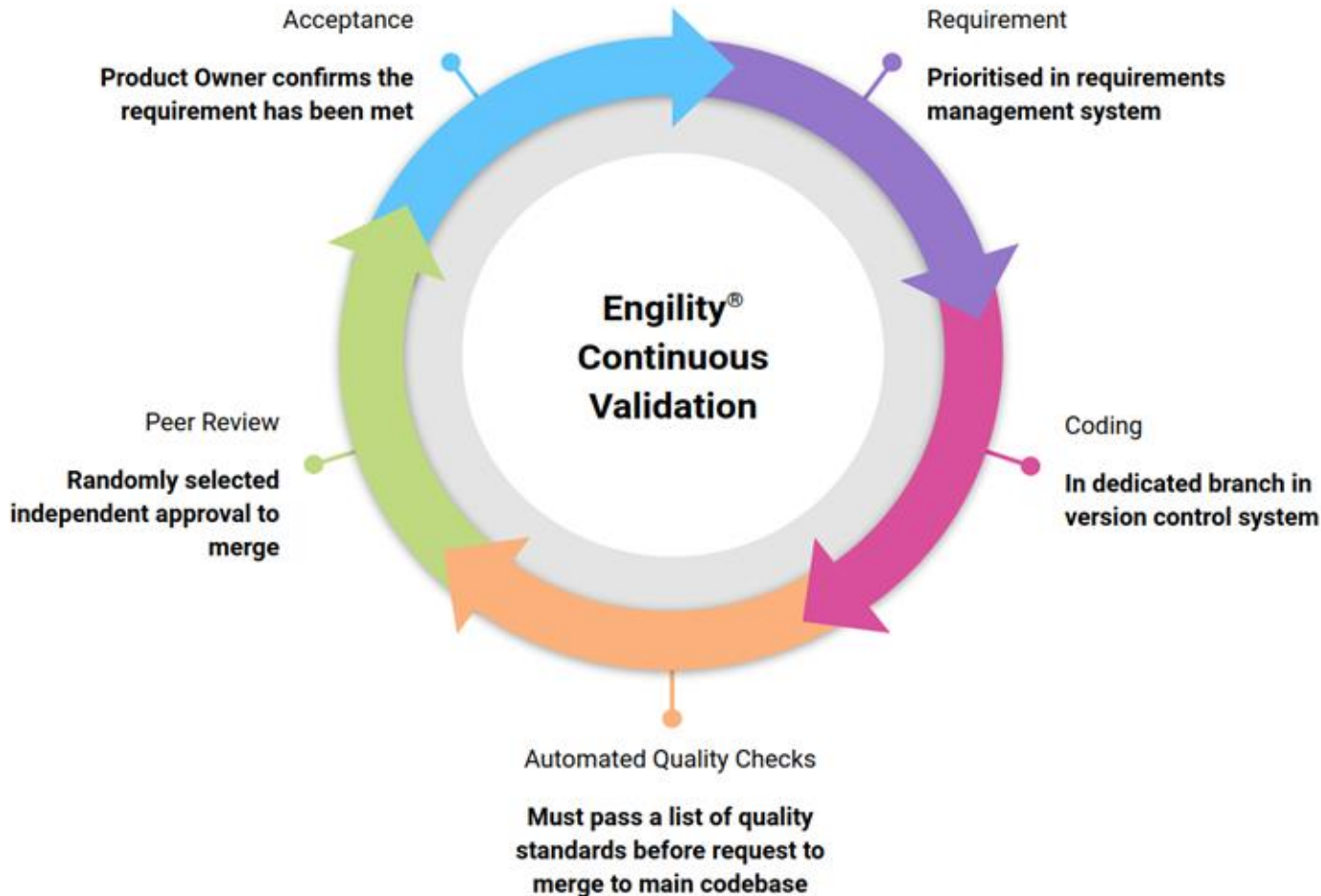
FEATURE	TREND: DEFECTS PER YEAR		TREND: DEFECTS PER YEAR		
	TOTAL	PERCENT	2022	2023	2024
A	136	43%	70	39	27
B	13	4%	4	8	1
C	15	4%	9	5	1
D	11	4%	7	3	1
E	20	6%	10	10	0
F	59	19%	36	18	5

Case Study: Pharmseal's Measurement Journey

Modern Software Development



Continuous Validation – Example



- Keep work chunks small – 1-5 days
- Work separately from main development branch
- Configuration management integrated into source code
- Requirement record linked to code change events
- All QC and testing completed immediately after development delivered
- New work merged into main development branch when QC and peer review passed
- Use automation for all repetitive tasks:
 - Automated build verification
 - Automated testing
 - Coding standard enforcement
 - Security vulnerabilities
- If story rejected, then return to start of process

Validation - in summary...

For each change...

- Automated process:
 - Build system from scratch
 - Full test suite run
 - Security vulnerability static testing
 - Coding standards checks
- Manual process:
 - Acceptance testing

For each release...

- Management approval of content
- Freeze code
- Confirmation automated test run:
 - Full test suite
 - Security vulnerability static testing
- Release to Staging environment
- Customer evaluation
- Release to Production environment

Improving Quality...

Audits – confirm that the process is being followed:

- Testing written for each new function
- Functional testing passed
- Static testing passed
- Peer review completed
- Acceptance testing completed

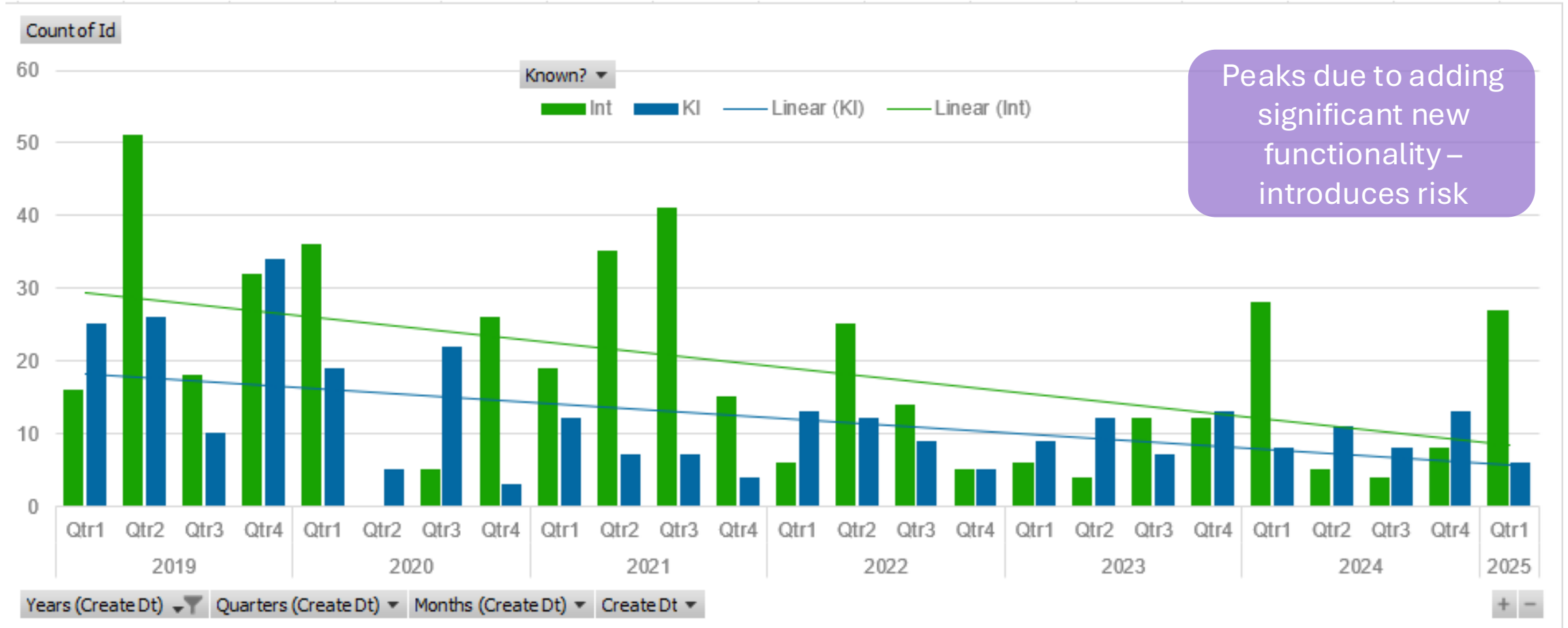
But, does the process work?

Measure defects

- Rejections – requirement gaps, missed functionality, bugs, design, ...
- Bugs identified after acceptance
 - Before release
 - After release
- Hotfixes – unplanned releases to correct bugs

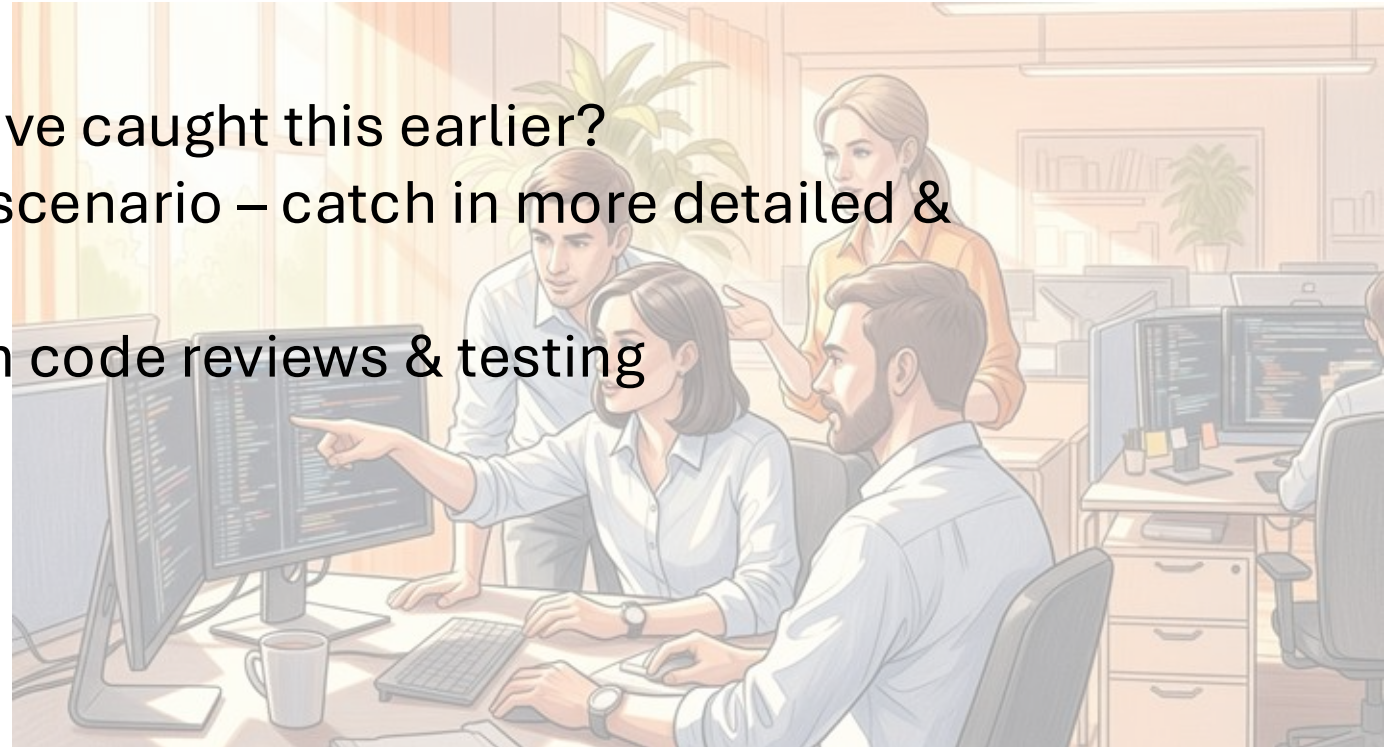


Continuous Improvement



Bugs – Root Cause Analysis

- For each bug, two new fields
 - Classify reason
 - Textual description
- Is there a way we could have caught this earlier?
 - Design flaws? – missed scenario – catch in more detailed & rigorous design reviews
 - Coding errors? – catch in code reviews & testing



Continuous Improvement



Where we started

2016 – Cont. Delivery
PR Checklist
Added Brakeman
Test strategy (improve performance)



Where we are

Bug cause data
Root cause analysis

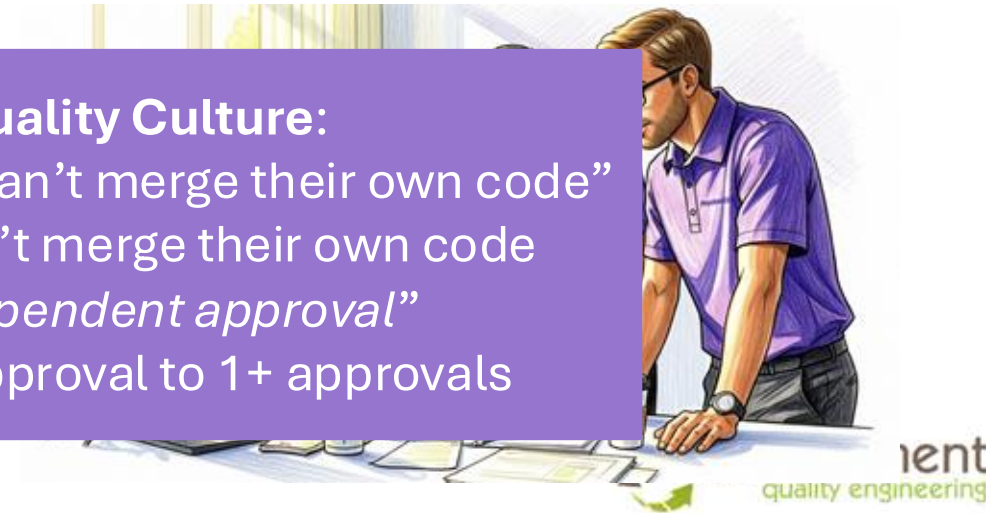


Our goal

Reduce bug impact
Reduce hotfixes

Team Quality Culture:

Change “developer can’t merge their own code”
to “developer can’t merge their own code
without independent approval”
– move from 1 approval to 1+ approvals



Returning to the SLC processes...

Where is PHARMASEAL on this model?



Advanced IT Auditing: Summary

The Challenge: Documentation vs. Reality:

- 88% of suppliers lack defined SLC technical process steps despite QMS documentation compliance. (2 Tier QMS?)

Advanced IT Auditing Techniques:

- **Technical Walkthroughs:** Deep-dive reviews revealing gaps in requirements, design, risk assessments, code, unit tests & code reviews. (Need technical & QA expertise).
- **Secondary Information Sources:** Analysis of SOC2 reports, wikis, and informal artefacts. (Need to know where to look).
- **Software Quality Analytics:** Leveraging defect data to assess both product quality and QMS effectiveness.
 - For example: Defect Removal Efficiency (DRE), Defects per release/feature, Root Cause Analysis, Defect Cost & Effort. (Just need to start trending).

Regulatory & Standards Foundation:

- Supported by EMA GVP, EU Annex 11, FDA, IEEE 1061, and ISO 9001 (1994-2015), GMLP (2021), Annex 22 (draft), ISO 5259
- The *core* IT QMS management technique | enables continuous improvement.

Thank You

Want to find our more: Contact Barry & Hugh on LinkedIn...